Amendments to the Drawings:

The attached replacement sheets of drawings include changes to Figures 1–2, 4, and 6–7. Each replacement sheet of drawings includes all of the figures appearing on the immediate prior version of the sheet.

Attachment: Replacement Sheets (4)

REMARKS

Applicants respectfully traverse and request reconsideration.

By this amendment and response, Applicants have amended the written description, claims 1–2, 4, 8–9, 11, 14–17, 19–21, and 24–25, and Figures 1–2, 4, and 6–7 to address typographical errors and to provide antecedent basis for claim terms. Claim 22 is cancelled without prejudice. With respect to the amended figures, reference numeral 128 has been added and arrow 120 pointing to arbiter 101 has been replaced with reference numeral 134 in Figure 1, the label "RESERVATION STATION" has been repositioned in Figure 2, reference numerals 332 and 334 have been added to Figure 4, the term "writing" has been added after the conjunction "and" in each of method blocks 408, 436, and 438 in Figures 6–7, and the preposition "to" as been added after the word "thread" and before the second instance of the article "the" in block 430 of Figure 7. Applicants respectfully submit that the aforementioned amendments are intended to clarify the written description, claims and figures and do not add new subject matter not previously presented in the originally filed application. Applicants further submit that the amendments to the claims do not narrow the literal scope of the claims or substantially relate to patentability for at least the reason that the above claim amendments do not address specific 35 U.S.C. § 112 rejections and were not necessary in view of the cited prior art as they are believed to correct typographical errors, provide proper antecedent basis or otherwise be cosmetic in nature.

Claims 1–4, 6–9, and 11–13 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,650,330 to Lindholm et al. ("Lindholm '330"). Lindholm '330 is directed to a graphics system and method of executing multiple threads in a transform module. In operation, the method includes determining a current thread to be executed and using a corresponding program counter number from a program counter to retrieve an instruction associated with the

current thread. (Col. 10, ll. 17–32; FIG. 4A). Lindholm '330 appears to teach that each execution thread has an associated program counter that is used to access instructions, or code segments, in a single instruction memory. (Col. 11, ll. 7–10). In other words, the single instruction memory stores execution threads while the plurality of program counters, each affiliated with an execution thread, is used to access the instructions or code segments from the single instruction memory. The instruction is then executed on a graphics-processing module such as an adder or multiplier. (Col. 10, ll. 32–34; Col. 11, ll. 10–12). "After the execution of each instruction, the program counter of the current execution thread is updated and the program counter of the next execution thread is called by module 458 in a round robin sequence to access an associated instruction." (Col. 11, ll. 25–30). Noticeably, threads in the instruction memory are not updated after execution of a current thread.

Claim 1 requires, among other things, a command processing engine that performs at least one processing command from the first command thread and thereupon updates the first command thread in the reservation station. As claimed, the reservation station has a plurality of command threads stored therein. The Office Action states that Lindholm '330 allegedly anticipates claim 1 because the program counter is updated after execution of a current thread. However, as discussed above, the program counter does not store a plurality of command threads. In contrast, Lindholm '330's single instruction memory stores execution threads while each program counter and its corresponding program counter number is used to retrieve a corresponding instruction or code segment from the instruction memory. Thus, Applicants submit that Lindholm '330 fails to teach or suggest updating the first command thread in the reservation station. For at least this reason, claim 1 is believed to be in proper condition for allowance.

Claims 2–4 and 6–7 depend upon allowable claim 1 and further add additional novel and non-obvious subject matter. For at least this reason, claims 2–4 and 6–7 are also believed to be in proper condition for allowance.

Claim 8 requires, among other things, a first reservation station having a plurality of first command threads stored therein and a second reservation station having a plurality of graphic command threads stored therein. The Office Action states that because "an instruction associated with a thread to be executed during a current cycle is retrieved using a corresponding program counter number," this implies a reservation station having a plurality of command threads stored therein. The Office Action further suggests that the "second program counter is associated with a second execution thread" and therefore implies the disclosure of the claimed second reservation station. However, Applicants respectfully submit that, at best, the cited portion of Lindholm '330 appears to teach a single instruction memory containing all execution threads as indicated in at least FIG. 4B of the reference. "As shown, each execution thread has an associated program counter that is used to access instructions, or code segments, in the instruction memory." (Col. 11, ll. 7–10). Because only a single instruction memory containing all execution threads is taught, the cited reference fails to disclosed Applicants' claimed first reservation station having a plurality of first command threads stored therein and the claimed second reservation station having a plurality of graphic command threads stored therein. For at least this reason, claim 8 is believed to be in proper condition for allowance.

Claim 8 further requires "an arbiter coupled to the first reservation station and the second reservation station such that the arbiter retrieves a selected command thread from one of the plurality of first command threads and the plurality of graphic command threads." The Office Action states that because "an instruction associated with a thread to be executed during a

current cycle is retrieved using a corresponding program counter number" is disclosed in Lindholm '330, the claimed arbiter is taught. In other words, it appears that the Office Action is associating one of the disclosed program counters with Applicants' claimed arbiter. Applicants, however, respectfully note that this is improper for at least the reason that Lindholm '330 teaches multiple program counters where each program counter is coupled to the same instruction memory and where each program counter is capable of accessing a single execution thread stored in the instruction memory. Lindholm '330 does not teach or suggest an arbiter coupled to both a first and second reservation station "such that the arbiter retrieves a selected command thread from one of the plurality of first command threads and the plurality of graphic command threads." For these reasons, claim 8 is believed to be in proper condition for allowance.

Claims 9 and 11–13 are dependent upon allowable claim 8 and further contain additional novel and nonobvious subject matter not disclosed in Lindholm '330. For at least the reasons articulated above with respect to claim 8, claims 9 and 11–13 are also believed to be allowable over the cited prior art.

Claims 14–25 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 7,015,913 to Lindholm et al. ("Lindholm '913"). Lindholm '913 is directed toward a method and apparatus for multithreaded processing of data in a programmable graphics processor. The programmable graphics processor includes a programmable graphics processing pipeline comprising a vertex input buffer 220 that receives and stores surfaces or primitives and a pixel input buffer 215 that receives and stores pixels or fragments. (Col. 4, ll. 35–40; Col. 5, ll. 20–25). The vertex input buffer 220 and pixel input buffer 215 are operatively coupled to at least one execution pipeline 240, each of which include at least one multithreaded processing unit 400. (Col. 4, ll. 40–55). The execution pipelines 240 are configured to receive a surface, primitive,

fragment or pixel (i.e., a sample) from, for example, the vertex input buffer 220 or the pixel input buffer 215 and to process the samples using an execution unit. (Col. 4, ll. 62–67). Execution pipelines 240 are further configured to communicate with texture unit 225 "using a read interface … to read program instructions and graphics data such as texture maps from local memory 140 or host memory 112 via memory controller 120 and a texture cache 230." (Col. 4, ll. 62–67). The samples received by each execution pipeline 240 are subsequently operated upon by an execution unit 470 as specified by these program instructions. (Col. 9, ll. 40–50). In other words, the program instructions read by texture unit 225 are used to configure each execution pipeline 240. (Col. 5, ll. 45–52).

Each execution pipeline 240 includes an instruction cache 410 that is operative to read thread entries from a thread control buffer 220 and to determine if requested program instructions are available in the instruction cache 410. (Col. 7, l. 59 to Col. 8, l. 14). "When a requested program instruction is not available in the instruction cache 410, it is read (possibly along with other program instructions stored in adjacent memory locations) from graphics memory," (Col. 8, ll. 14–20) "i.e., host memory 112, local memory 140, register files coupled to the components within graphics processor 105, and the like" (Col. 6, ll. 14–18) using the texture unit 225 (Fig. 4, element 410 and arrows to and from 225). Thereafter, an instruction scheduler 430 determines which instructions and associated threads will be executed by the execution unit 470.

Claim 14 requires, among other things, "a pixel reservation station having a plurality of pixel command threads stored therein" and "a vertex reservation station having a plurality of vertex commands stored therein." The Office Action asserts that the pixel input buffer 215 of Lindholm '913 is equivalent to the claimed pixel reservation station and that the vertex input

buffer 220 of Lindholm '913 is equivalent to the claimed vertex reservation station. However, Applicants respectfully submit that neither the pixel input buffer 215 nor the vertex input buffer 225 store a plurality of pixel command threads and a plurality of vertex command threads, respectively, as required by the claims. As discussed above, the pixel input buffer 215 stores pixels and fragments (col. 5, ll. 20–25) and the vertex input buffer 225 stores surfaces and primitives (col. 4, ll. 35–40). Because the pixel input buffer 215 does not have a plurality of pixel command threads stored therein and because the vertex input buffer 225 does not have a plurality of vertex commands stored therein, Lindholm '913 fails to teach or suggest each and every limitation presented in claim 14. For this reason alone, claim 14 appears to be in proper condition for allowance.

Because the cited prior art fails to teach each and every limitation presented in claim 14, Applicants respectfully believes claim 14 is in proper condition for allowance.

Claims 15–19 depend upon allowable claim 14 and further add additional novel and nonobvious subject matter. For at least the reasons claim 14 is allowed, Applicants respectfully believe claims 15–19 are also allowable.

As to claim 20, Applicants respectfully reasserts the relevant remarks above with respect to claim 14 and further notes that claim 20 requires "writing the selected command thread to a first reservation station." The Office Action suggests that the pixel output buffer 270 reads on this limitation. However, Applicants respectfully submits that the pixel output buffer 270, as defined above, is limited to storing fragments and pixels. (Col. 5, ll. 7–30). Because a selected command thread may not be written to the pixel output buffer 270 as claimed, Applicants respectfully submit that the present Office Action has failed to identify prior art that reads on

each and every limitation presented in the claim. For at least this reason, claim 20 appears to be in proper condition for allowance.

Claims 21–25 depend upon allowable claim 20 and further contain additional novel and nonobvious subject matter. For at least the reason claim 20 is allowed, claims 21–25 are also believed to be properly allowable over Lindholm '913.

Claims 5 and 10 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Lindholm '330 in view of Lindholm '913. Claims 5 and 10 are dependent upon allowable claims 1 and 8 and contain additional novel and nonobvious subject matter not present in the cited prior art. For at least the reasons claims 1 and 8 are allowable, Applicants respectfully believed claims 5 and 10 to be properly allowable.

Applicants respectfully submit that the claims are in condition for allowance and respectfully request that a timely Notice of Allowance be issued in this case. The Examiner is invited to contact the below listed attorney if the Examiner believes that a telephone conference will advance the prosecution of this application.

Respectfully submitted,

Date: 7/10/06

By: /Christopher J. Reckamp/
Christopher J. Reckamp
Registration No. 34,414

Vedder, Price, Kaufman & Kammholz, P.C.
222 North LaSalle Street, Suite 2600
Chicago, Illinois 60601
phone: (312) 609-7599
fax: (312) 609-5005